LISP Map Server Interface
draft-ietf-lisp-ms-12.txt

Abstract

   This draft describes the Maping Service for the Locator Identifier
   Separation Protocol (LISP), implemented by two new types of LISP-
   speaking devices, the LISP Map Resolver and LISP Map Server, that
   provides a simplified "front end" to for one or more Endpoint ID to
   Routing Locator mapping databases.

   By using this service interface and communicating with Map Resolvers
   and Map Servers, LISP Ingress Tunnel Routers and Egress Tunnel
   Routers, are not dependent on the details of mapping database
   systems, which facilitates experimentation with different database
   designs. Since these devices implement the "edge" of the LISP
   infrastructure, connect directly to LISP-capable Internet end sites,
   and comprise the bulk of LISP-speaking devices, reducing their
   implementation and operational complexity should also reduce the
   overall cost and effort of deploying LISP.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

[LISP], the Locator Identifier Separation Protocol, specifies an
architecture and mechanism for replacing the addresses currently used
by IP with two separate name spaces: Endpoint IDs (EIDs), used within
sites, and Routing Locators (RLOCs), used on the transit networks
that make up the Internet infrastructure. To achieve this
separation, LISP defines protocol mechanisms for mapping from EIDs to
RLOCs. In addition, LISP assumes the existence of a database to
store and propagate those mappings globally. Several such databases
have been proposed, among them: LISP-CONS [CONS], LISP-NERD, [NERD]
and LISP+ALT [ALT].

The LISP Mapping Service defines two new types of LISP-speaking
devices: the Map Resolver, which accepts Map-Requests from an Ingress
Tunnel Router (ITR) and "resolves" the EID-to-RLOC mapping using a
mapping database, and the Map Server, which learns authoritative EID-
to-RLOC mappings from an Egress Tunnel Router (ETR) and publishes
them in a database.

Conceptually, LISP Map Servers share some of the same basic
configuration and maintenance properties as Domain Name System (DNS)
[RFC1035] servers; likewise, Map Resolvers are conceptually similar
to DNS caching resolvers. With this in mind, this specification
borrows familiar terminology (resolver and server) from the DNS
specifications.

Note that while this document assumes a LISP+ALT database mapping
infrastructure to illustrate certain aspects of Map Server and Map
Resolver operation, the Mapping Service interface can (and likely
will) be used by ITRs and ETRs to access other mapping database
systems as the LISP infrastructure evolves.

2. Definition of Terms

   Map Server: a network infrastructure component which learns of EID-
      prefix mapping entries from an ETR, via the registration mechanism
      described below, or some other authoritative source if one exists.
      A Map Server publishes these EID-prefixes in a mapping database.

   Map Resolver: a network infrastructure component which accepts LISP
      Encapsulated Map-Requests, typically from an ITR, determines
      whether or not the destination IP address is part of the EID
      namespace; if it is not, a Negative Map-Reply is returned.
      Otherwise, the Map Resolver finds the appropriate EID-to-RLOC
      mapping by consulting a mapping database system.

   Encapsulated Map-Request: a LISP Map-Request carried within an
      Encapsulated Control Message, which has an additional LISP header
      prepended. Sent to UDP destination port 4342. The "outer"
      addresses are globally-routeable IP addresses, also known as
      RLOCs. Used by an ITR when sending to a Map Resolver and by a Map
      Server when forwarding a Map-Request to an ETR.

   Negative Map-Reply: a LISP Map-Reply that contains an empty
      locator-set. Returned in response to a Map-Request if the
      destination EID does not exist in the mapping database.
      Typically, this means that the "EID" being requested is an IP
      address connected to a non-LISP site.

   Map-Register message: a LISP message sent by an ETR to a Map Server
      to register its associated EID-prefixes. In addition to the set
      of EID-prefixes to register, the message includes one or more
      RLOCs to be be used by the Map Server when forwarding Map-Requests
      (re-formatted as Encapsulated Map-Requests) received through the
      database mapping system. An ETR may request that the Map Server
      answer Map-Requests on its behalf by setting the "proxy-map-reply"
      flag (P-bit) in the message.

   Map-Notify message: a LISP message sent by a Map Server to an ETR
      to confirm that a Map-Register has been received and processed.
      An ETR requests that a Map-Notify be returned by setting the
      "want-map-notify" or "M" bit in the Map-Register message. Unlike
      a Map-Reply, a Map-Notify uses UDP port 4342 for both source and
      destination.

   For definitions of other terms, notably Map-Request, Map-Reply,
   Ingress Tunnel Router (ITR), and Egress Tunnel Router (ETR), please
   consult the LISP specification [LISP].

3. **Basic Overview**

   A Map Server is a device which publishes EID-prefixes in a LISP
   mapping database on behalf of a set of ETRs. When it receives a Map
   Request (typically from an ITR) it consults the mapping database to
   find an ETR that can answer with the set of RLOCs for an EID-prefix.
   To publish its EID-prefixes, an ETR periodically sends Map-Register
   messages to the Map Server. A Map-Register message contains a list
   of EID-prefixes plus a set of RLOCs that can be used to reach the ETR
   when a Map Server needs to forward a Map-Request to it.

   When LISP+ALT is used as the mapping database, a Map Server connects
   to ALT network and acts as a "last-hop" ALT router. Intermediate ALT
   routers forward Map-Requests to the Map Server that advertises a
   particular EID-prefix and the Map Server forwards them to the owning
   ETR, which responds with Map-Reply messages.

   A Map Resolver receives Encapsulated Map-Requests from its client
   ITRs and uses a mapping database system to find the appropriate ETR
   to answer those requests. On a LISP+ALT network, a Map Resolver acts
   as a "first-hop" ALT router. It has GRE tunnels configured to other
   ALT routers and uses BGP to learn paths to ETRs for different
   prefixes in the LISP+ALT database. The Map Resolver uses this path
   information to forward Map-Requests over the ALT to the correct ETRs.
   During initial deployment, a Map Resolver will operate only in a non-
   caching mode, where it simply de-capsulates and forwards the
   Encapsulated Map-Requests that it receives from ITRs.

   In future deployments, a Map Resolver may operate in a caching mode,
   where it saves information about outstanding Map-Requests, originates
   new Map-Requests to the correct ETR(s), accepts and caches the Map-
   Replies, and finally forwards the Map-Replies to the original ITRs.
   One significant issue with use of caching in a Map Resolver is that
   it hides the original ITR source of a Map-Request, which prevents an
   ETR from tailoring its responses to that source; this reduces the
   inbound traffic- engineering capability for the site owning the ETR.
   In addition, caching in a Map Resolver exacerbates problems
   associated with old mappings being cached; an outdated, cached
   mapping in an ITR affects only that ITR and traffic originated by its
   site while an outdated, cached mapping in a Map Resolver could cause
   a problem with a wider scope. More experience with caching Map
   Resolvers on the LISP pilot network will be needed to determine
   whether their use can be recommended.

   Note that a single device can implement the functions of both a Map
   Server and a Map Resolver and, in many cases, the functions will be
   co-located in that way.

   Detailed descriptions of the LISP packet types referenced by this
   document may be found in [LISP].

4. Interactions With Other LISP Components

4.1. ITR EID-to-RLOC Mapping Resolution

   An ITR is configured with one or more Map Resolver addresses. These
   addresses are "locators" (or RLOCs) and must be routeable on the
   underlying core network; they must not need to be resolved through
   LISP EID-to-RLOC mapping as that would introduce a circular
   dependency. When using a Map Resolver, an ITR does not need to
   connect to any other database mapping system. In particular, the ITR
   need not connect to the LISP+ALT infrastructure or implement the BGP
   and GRE protocols that it uses.

   An ITR sends an Encapsulated Map-Request to a configured Map Resolver
   when it needs an EID-to-RLOC mapping that is not found in its local
   map-cache. Using the Map Resolver greatly reduces both the
   complexity of the ITR implementation and the costs associated with
   its operation.

   In response to an Encapsulated Map-Request, the ITR can expect one of
   the following:

   o An immediate Negative Map-Reply (with action code of "forward-
      native", 15-minute TTL) from the Map Resolver if the Map Resolver
      can determine that the requested EID does not exist. The ITR
      saves the EID-prefix returned in the Map-Reply in its cache,
      marking it as non-LISP-capable and knows not to attempt LISP
      encapsulation for destinations matching it.

   o A Negative Map-Reply (with action code of "forward-native") from
      the Map Server that has an aggregate EID-covering the EID in the
      Map-Request but where the EID matches a "hole" in the aggregate.
      If the "hole" is for a LISP EID-prefix that is defined in the Map
      Server configuration but for which no ETRs are currently
      registered, a 1-minute TTL is returned. If the "hole" is for an
      unassigned part of the aggregate, then it is not a LISP EID and a
      15-minute TTL is returned. See Section 4.2 for discussion of
      aggregate EID-prefixes and details of Map Server EID-prefix
      matching.

   o A LISP Map-Reply from the ETR that owns the EID-to-RLOC mapping or
      possibly from a Map Server answering on behalf of the ETR. See
      (Section 4.4) for more details on Map Resolver message processing.

   Note that an ITR may be configured to both use a Map Resolver and to
   participate in a LISP+ALT logical network. In such a situation, the
   ITR should send Map-Requests through the ALT network for any EID-
   prefix learned via ALT BGP. Such a configuration is expected to be

very rare, since there is little benefit to using a Map Resolver if
an ITR is already using LISP+ALT. There would be, for example, no
need for such an ITR to send a Map-Request to a possibly non-existent
EID (and rely on Negative Map-Replies) if it can consult the ALT
database to verify that an EID-prefix is present before sending that
Map-Request.

4.2. EID Prefix Configuration and ETR Registration

An ETR publishes its EID-prefixes on a Map Server by sending LISP
Map-Register messages. A Map-Register message includes
authentication data, so prior to sending a Map-Register message, the
ETR and Map Server must be configured with a shared secret or other
relevant authentication information. A Map Server's configuration
must also include a list of the EID-prefixes for which each ETR is
authoritative. Upon receipt of a Map-Register from an ETR, a Map
Server accepts only EID-prefixes that are configured for that ETR.
Failure to implement such a check would leave the mapping system
vulnerable to trivial EID-prefix hijacking attacks. As developers
and operators gain experience with the mapping system, additional,
stronger security measures may be added to the registration process.

In addition to the set of EID-prefixes defined for each ETR that may
register, a Map Server is typically also configured with one or more
aggregate prefixes that define the part of the EID numbering space
assigned to it. When LISP+ALT is the database in use, aggregate EID-
prefixes are implemented as discard routes and advertised into ALT
BGP. The existance of aggregate EID-prefixes in a Map Server's
database means that it may receive Map Requests for EID-prefixes that
match an aggregate but do not match a registered prefix; Section 4.3
describes how this is handled.

Map-Register messages are sent periodically from an ETR to a Map
Server with a suggested interval between messages of one minute. A
Map Server should time-out and remove an ETR's registration if it has
not received a valid Map-Register message within the past three
minutes. When first contacting a Map Server after restart or changes
to its EID-to-RLOC database mappings, an ETR may initially send Map-
Register messages at an increased frequency, up to one every 20
seconds. This "quick registration" period is limited to five minutes
in duration.

An ETR may request that a Map Server explicitly acknowledge receipt
and processing of a Map-Register message by setting the "want-map-
notify" ("M" bit) flag. A Map Server that receives a Map-Register
with this flag set will respond with a Map-Notify message. Typical
use of this flag by an ETR would be to set it for Map-Register
messages sent during the initial "quick registration" with a Map

Server but then set it only occasionally during steady-state
maintenance of its association with that Map Server. Note that the
Map-Notify message is sent to UDP destination port 4342, not to the
source port specified in the original Map-Register message.

Note that a one-minute minimum registration interval during
maintenance of an ETR-MS association places a lower-bound on how
quickly and how frequently a mapping database entry can be updated.
This may have implications for what sorts of mobility can be
supported directly by the mapping system; shorter registration
intervals or other mechanisms might be needed to suopport faster
mobility in some cases. For a discussion on one way that faster
mobility may be implemented for individual devices, please see
[LISP-MN].

An ETR may also request, by setting the "proxy-map-reply" flag
(P-bit) in the Map-Register message, that a Map Server answer Map-
Requests instead of forwarding them to the ETR. See [LISP] for
details on how the Map Server sets certain flags (such as those
indicating whether the message is authoritative and how returned
locators should be treated) when sending a Map-Reply on behalf of an
ETR. When an ETR requests proxy reply service, it should include all
RLOCs for all ETRs for the EID-prefix being registered, along with
the routable flag ("R-bit") setting for each RLOC. The Map Server
includes all of this information in Map Reply messages that it sends
on behalf of the ETR. This differs from a non-proxy registration
since the latter need only provide one or more RLOCs for a Map Server
to use for forwarding Map-Requests; the registration information is
not used in Map-Replies so it being incomplete is not incorrect.

An ETR which uses a Map Server to publish its EID-to-RLOC mappings
does not need to participate further in the mapping database
protocol(s). When using a LISP+ALT mapping database, for example,
this means that the ETR does not need to implement GRE or BGP, which
greatly simplifies its configuration and reduces its cost of
operation.

Note that use of a Map Server does not preclude an ETR from also
connecting to the mapping database (i.e. it could also connect to the
LISP+ALT network) but doing so doesn't seem particularly useful as
the whole purpose of using a Map Server is to avoid the complexity of
the mapping database protocols.

4.3. Map Server Processing

Once a Map Server has EID-prefixes registered by its client ETRs, it
can accept and process Map-Requests for them.

In response to a Map-Request (received over the ALT if LISP+ALT is in use), the Map Server first checks to see if the destination EID matches a configured EID-prefix. If there is no match, the Map Server returns a negative Map-Reply with action code "forward-native" and a 15-minute TTL. This may occur if a Map Request is received for a configured aggreate EID-prefix for which no more-specific EID-prefix exists; it indicates the presence of a non-LISP "hole" in the agregate EID-prefix.

Next, the Map Server checks to see if any ETRs have registered the matching EID-prefix. If none are found, then the Map Server returns a negative Map-Reply with action code "forward-native" and a 1-minute TTL.

If any of the registered ETRs for the EID-prefix have requested proxy reply service, then the Map Server answers the request instead of forwarding it. It returns a Map-Reply with the EID-prefix, RLOCs, and other information learned through the registration process.

If none of the ETRs have requested proxy reply service, then the Map Server re-encapsulates and forwards the resulting Encapsulated Map-Request to one of the registered ETRs. It does not otherwise alter the Map-Request so any Map-Reply sent by the ETR is returned to the RLOC in the Map-Request, not to the Map Server. Unless also acting as a Map Resolver, a Map Server should never receive Map-Replies; any such messages should be discarded without response, perhaps accompanied by logging of a diagnostic message if the rate of Map-Replies is suggestive of malicious traffic.

A Map Server may also receive a Map-Request that is contained inside of an Encapsulated Control Message (an Encapsulated Map-Request) with the "Security" bit ("S-bit") set. It processes the security parameters as described in [LISP-SEC] then handles the Map-Request as as described above.

Note that a Map Server that is sending a Map-Reply on behalf of an ETR (performing proxy reply service) must perform security processing for that ETR as well; see [LISP-SEC] for details.

4.4. Map Resolver Processing

Upon receipt of an Encapsulated Map-Request, a Map Resolver de-capsulates the enclosed message then searches for the requested EID in its local database of mapping entries (statically configured, cached, or learned from associated ETRs when the Map Resolver is also acting as a Map Server). If it finds a matching entry, it returns a non-authoritative LISP Map-Reply with the known mapping.

If the Map Resolver does not have the mapping entry and if it can
determine that the EID is not in the mapping database (for example,
if LISP+ALT is used, the Map Resolver will have an ALT forwarding
table that covers the full EID space) it immediately returns a
negative LISP Map-Reply, with action code "forward-native" and a 15-
minute TTL. To minimize the number of negative cache entries needed
by an ITR, the Map Resolver should return the least-specific prefix
which both matches the original query and does not match any EID-
prefix known to exist in the LISP-capable infrastructure.

If the Map Resolver does not have sufficient information to know
whether the EID exists, it needs to forward the Map-Request to
another device which has more information about the EID being
requested. This is done in one of two ways:

1. A non-caching Map Resolver simply forwards the unencapsulated
    Map-Request, with the original ITR RLOC as the source, to the
    mapping database system. Using LISP+ALT, the Map Resolver is
    connected to the ALT network and sends the Map-Request to the
    next ALT hop learned from its ALT BGP neighbors. The Map
    Resolver does not send any response to the ITR; since the source
    RLOC is that of the ITR, the ETR or Map Server which receives the
    Map-Request over the ALT and responds will do so directly to the
    ITR.

2. A caching Map Resolver queues information from the Encapsulated
    Map-Request, including the ITR RLOC and the original Map-Request
    message nonce. It then modifies the Map-Request to use its own
    RLOC, generates a "local nonce" (which is also saved in the
    request queue entry), and forwards the Map-Request as above.
    When the Map Resolver receives a Map-Reply, it looks in its
    request queue to match the reply nonce to a "local nonce" entry
    then de-queues the entry and uses the saved original nonce and
    ITR RLOC to re-write those fields in the Map-Reply before sending
    to the ITR. The request queue entry is also deleted and the
    mapping entries from the Map-Reply are saved in the Map
    Resolver's cache.

If a Map Resolver receives a Map-Request contained in an Encapsulated
Control Message (an Encapsulated Map-Request) with the "security"
option ("S-Bit") set, additional processing is required. It extracts
the enclosed Map-Request and uses the attached security paramaters to
generate a new Encapsulated Control Message containing the original
Map-Request and additional signature information used to protect both
the Map-Request and the Map-Reply that will be generated by the
destination ETR or Map Server. The outgoing message will have the
"S-bit" set, will use the requested EID as its outer header
destination IP address plus Map Resolver RLOC as source IP address,

and will include security parameters added by the Map Resolver. See
[LISP-SEC] for details of the checks that are performed and the
security information that is added during the de-encapsulation and
re-encapsulation process.

4.4.1. Anycast Map Resolver Operation

A Map Resolver can be set up to use "anycast", where the same address
is assigned to multiple Map Resolvers and is propagated through IGP
routing, to facilitate the use of a topologically-close Map Resolver
each ITR.

Note that Map Server associations with ETRs should not use anycast
addresses as registrations need to be established between an ETR and
a specific set of Map Servers, each identified by a specific
registation association.

5. Open Issues and Considerations

   There are a number of issues with the Map Server and Map Resolver
   design that are not yet completely understood. Among these are:

   o Constants, such as those used for Map-Register frequency,
      retransmission timeouts, retransmission limits, negative Map-Reply
      TTLs, et al are subject to further refinement as more experience
      with prototype deployment is gained.

   o Feasibility, performance, and complexity trade-offs of
      implementing caching in Map Resolvers, as discussed in Section 3,
      Paragraph 4.

   o Convergence time when an EID-to-RLOC mapping changes and
      mechanisms for detecting and refreshing or removing stale, cached
      information

   o Deployability and complexity trade-offs of implementing stronger
      security measures in both EID-prefix registration and Map-Request/
      Map-Reply processing

   o Requirements for additional state in the registration process
      between Map Servers and ETRs

   The authors expect that experimentation on the LISP pilot network
   will help answer open questions surrounding these and other issues.

6. IANA Considerations

This document makes no request of the IANA.

7. Security Considerations

The 2-way LISP header nonce exchange documented in [LISP] can be used to avoid ITR spoofing attacks.

To publish an authoritative EID-to-RLOC mapping with a Map Server, an ETR includes authentication data that is a hash of the message using pair-wise shared key. An implementation must support use of HMAC-SHA-1-160 [RFC2104] and should support use of HMAC-SHA-256-128 [RFC6234] (SHA-256 truncated to 128 bits).

During experimental and prototype deployment, all authentication key configuration will be manual. Should LISP and its components be considered for IETF standardization, further work will be required to follow the BCP 107 [RFC4107] recommendations on automated key management.

As noted in Section 4.2, a Map Server should verify that all EID-prefixes registered by an ETR match configuration stored on the Map Server.

[LISP-SEC] defines a mechanism for providing origin authentication, integrity, anti-reply protection, and prevention of man-in-the-middle and "overclaiming" attacks on the Map-Request/Map-Reply exchange.

While beyond the scope of securing an individual Map Server or Map Resolver, it should be noted that a BGP-based LISP+ALT network (if ALT is used as the mapping database infrastructure) can take advantage of technology being developed by the IETF SIDR working group or either S-BGP [I-D.murphy-bgp-secr] or soBGP [I-D.white-sobgparchitecture] should they be developed and widely deployed.

8. References

8.1. Normative References

   [ALT]   Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "LISP
              Alternative Topology (LISP-ALT)",
              draft-ietf-lisp-alt-10.txt (work in progress),
              October 2011.

   [LISP]   Farinacci, D., Fuller, V., Meyer, D., and D. Lewis,
              "Locator/ID Separation Protocol (LISP)",
              draft-ietf-lisp-15.txt (work in progress), July 2011.

   [LISP-SEC]
              Maino, F., Ermagan, V., Cabellos, A., Sanchez, D., and O.
              Bonaventure, "LISP-Security", draft-ietf-lisp-sec-00.txt
              (work in progress), July 2011.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, November 1987.

   [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-
              Hashing for Message Authentication", RFC 2104,
              February 1997.

   [RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms
              (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

8.2. Informative References

   [CONS]   Farinacci, D., Fuller, V., and D. Meyer, "LISP-CONS: A
              Content distribution Overlay Network Service for LISP",
              draft-meyer-lisp-cons-04.txt (work in progress),
              April 2008.

   [I-D.murphy-bgp-secr]
              Murphy, S., "BGP Security Analysis",
              draft-murphy-bgp-secr-04 (work in progress),
              November 2001.

   [I-D.white-sobgparchitecture]
              White, R., "Architecture and Deployment Considerations for
              Secure Origin BGP (soBGP)",
              draft-white-sobgparchitecture-00 (work in progress),
              May 2004.

   [LISP-MN] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "LISP
              Mobile Node Architecture", draft-meyer-lisp-mn-05.txt

                  (work in progress), May 2011.

   [NERD]   Lear, E., "NERD: A Not-so-novel EID to RLOC Database",
                  draft-lear-lisp-nerd-08.txt (work in progress),
                  March 2010.

   [RFC4107]   Bellovin, S. and R. Housley, "Guidelines for Cryptographic
                  Key Management", BCP 107, RFC 4107, June 2005.

Appendix A. Acknowledgments

   The authors would like to thank Greg Schudel, Darrel Lewis, John
   Zwiebel, Andrew Partan, Dave Meyer, Isidor Kouvelas, Jesper Skriver,
   Fabio Maino, and members of the lisp@ietf.org mailing list for their
   feedback and helpful suggestions.

   Special thanks are due to Noel Chiappa for his extensive work on
   caching with LISP-CONS, some of which may be used by Map Resolvers.

Authors' Addresses

   Vince Fuller
   cisco Systems
   Tasman Drive
   San Jose, CA 95134
   USA

   Email: vaf@cisco.com

   Dino   Farinacci
   cisco Systems
   Tasman Drive
   San Jose, CA 95134
   USA

   Email: dino@cisco.com